

Parameterized Algorithmics for Computational Social Choice: Nine Research Challenges

Robert Brederbeck¹, Jiehua Chen¹, Piotr Faliszewski², Jiong Guo³, Rolf Niedermeier¹, and Gerhard J. Woeginger⁴

¹Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany,
 robert.bredereck@tu-berlin.de, jiehua.chen@tu-berlin.de, rolf.niedermeier@tu-berlin.de

²AGH University of Science and Technology, Krakow, Poland, faliszew@agh.edu.pl

³Cluster of Excellence Multimodal Computing and Interaction, Universität des Saarlandes,
 Saarbrücken, Germany, jguo@mmci.uni-saarland.de

⁴Department of Mathematics and Computer Science, TU Eindhoven, Eindhoven, The Netherlands,
 gwoegi@win.tue.nl

Abstract

Computational Social Choice is an interdisciplinary research area involving Economics, Political Science, and Social Science on the one side, and Mathematics and Computer Science (including Artificial Intelligence and Multiagent Systems) on the other side. Typical computational problems studied in this field include the vulnerability of voting procedures against attacks, or preference aggregation in multi-agent systems. Parameterized Algorithmics is a subfield of Theoretical Computer Science seeking to exploit meaningful problem-specific parameters in order to identify tractable special cases of in general computationally hard problems. In this paper, we propose nine of our favorite research challenges concerning the parameterized complexity of problems appearing in this context.

1 Introduction

Computational social choice [28, 39, 40, 4, 5, 58, 91] is a relatively young interdisciplinary research area that brings together researchers from fields like Artificial Intelligence, Decision Theory, Discrete Mathematics, Mathematical Economics, Operations Research, Political Sciences, Social Choice, and Theoretical Computer Science. The main objective is to improve our understanding of social choice mechanisms and of algorithmic decision making. Some concrete questions belonging to this area are: How should a voter choose between competing political alternatives? How should this voter rank these competing alternatives? How should a meta-search engine aggregate many rankings into a single consensus ranking, reflecting an “optimal” compromise between various alternatives? A long term goal is to improve decision support for decision makers that are working in a variety of areas like electronic commerce, logistics, recommender systems, risk assessment, risk management, and that are fighting with massive data sets, large combinatorial structures, and uncertain information.

The general topic is addressed by a biennial *International Workshop on Computational Social Choice (COMSOC)*, whose 2012 and 2014 editions have been held in Kraków/Poland and in Pittsburgh/Pennsylvania/USA, respectively. Furthermore, the topic is covered by a number of leading conferences in Artificial Intelligence (including *AAAI*, *ECAI*, *IJCAI*) and by several specialized conferences (including *AAMAS*, *ADT*, *EC*, *SAGT*, *WINE*). There are numerous research journals that address many aspects of computational social choice, including *Artificial Intelligence*, *ACM Transactions on Economics and Computation*, *Autonomous Agents and Multi-Agent Systems*, *Journal of Artificial Intelligence Research*, *Mathematical Social Sciences*, *Social Choice and Welfare*, and many others (including theoretical as well as application-oriented journals).

Parameterized Complexity is a branch of Theoretical Computer Science that started in the late 1980s and early 1990s. The main objective is to understand computational problems with respect to multiple input parameters, and to classify these problems according to their inherent difficulty. As the complexity of a problem can be measured as a function of a multitude of input parameters, this approach allows to classify NP-hard problems on a much finer scale than in classical complexity theory (where the complexity of a problem is only measured with respect to the number of bits in the input). Parameterized complexity analysis has been successfully applied in areas as diverse as Algorithm Engineering, Cognitive Sciences, Computational Biology, Computational Geometry, Geographic Information Systems, Machine Learning, and Psychology, to name a few. The first systematic work on Parameterized Complexity is the 1999 book [47] by Downey and Fellows.

This article suggests future research directions and open problems in the intersection area between computational social choice and parameterized complexity. Parts of this intersection area have been surveyed in a recent paper on the parameterized complexity analysis of voting problems [11]. The field holds numerous exciting challenges for researchers on algorithms and complexity.

2 Preliminaries

We summarize some basic concepts and definitions that will be used throughout this paper.

2.1 Voting

Elections. An election $E := (C, V)$ consists of a set C of m alternatives c_1, c_2, \dots, c_m and a list V of n voters v_1, v_2, \dots, v_n . Each voter v has a linear order \succ_v over the set C which we call a *preference order*. For example, let $C = \{c_1, c_2, c_3\}$ be a set of alternatives. The preference order $c_1 \succ_v c_2 \succ_v c_3$ of voter v indicates that v likes c_1 most (the 1st position), then c_2 , and c_3 least (the 3rd position). For any two distinct alternatives c and c' , we write $c \succ_v c'$ if voter v prefers c over c' . We also use the notation $v \in V$ to indicate that a voter v is in the list V . By $|V|$, we mean the number of voters in V .

Voting rules. A voting rule \mathcal{R} is a function that maps an election to a subset of alternatives, the set of winners. One prominent voting rule dates back to de Condorcet [43]. It selects an alternative as a (unique) winner if it beats any other alternative in head-to-head contests.

Formally speaking, given an election $E = (C, V)$, an alternative $c \in C$ is a *Condorcet winner* if any other alternative $c' \in C \setminus \{c\}$ satisfies

$$|\{v \in V \mid c \succ_v c'\}| > |\{v \in V \mid c' \succ_v c\}|.$$

It is easy to see that there are elections for which no alternative is a Condorcet winner. However, if a voting rule guarantees that a Condorcet winner is elected whenever there is one, then we say that this rule is Condorcet-consistent. (Such rules include, e.g., those of Copeland [68, 25], of Dodgson [45], of Kemeny [75, 79], and many others.)

Scoring protocols form another well-studied class of voting rules. A scoring protocol for m alternatives is defined through a vector $(\alpha_1, \dots, \alpha_m)$ of integers, $\alpha_1 \geq \dots \geq \alpha_m \geq 0$. An alternative receives α_i points for each voter that ranks this alternative as i th best. The best known examples of (families of) scoring rules include the Plurality rule (defined through a vector $(1, 0, \dots, 0)$), d -Approval (defined through vectors of d ones followed by zeros), and the Borda rule (defined through vectors $(m-1, m-2, \dots, 1, 0)$).

2.2 Parameterized Complexity

We assume familiarity with basic Computational Complexity Theory [3, 67] and only provide some definitions with respect to parameterized complexity analysis [48, 65, 87].

Let Σ be an alphabet. A *parameterized problem* over Σ is a *language* $L \subseteq \Sigma^* \times \Sigma^*$. The second part of the problem is called the *parameter*. Usually, this parameter is a non-negative integer or a tuple of non-negative integers. For instance, an obvious parameter in voting is the number of alternatives. Thus, typically $L \subseteq \Sigma^* \times \mathbb{N}$, where a combined parameter can be interpreted as the maximum of its integer components.

In the following, let L be a parameterized problem.

Fixed-parameter tractability. We say that L is *fixed-parameter tractable* or in FPT if there is an algorithm that, given an input (I, k) , decides in $f(k) \cdot |I|^{O(1)}$ time whether $(I, k) \in L$. Herein $f: \mathbb{N} \rightarrow \mathbb{N}$ is a computable function depending only on k .

Kernelization is an alternative way of showing fixed-parameter tractability [22, 70, 84]. We say that L has a *problem kernel* if there is a polynomial-time algorithm (that is, a kernelization) that given an instance (I, k) , computes an equivalent instance (I', k') whose size is upper-bounded by a function in k , that is,

- (i) (I, k) is a yes-instance if and only if (I', k') is a yes-instance, and
- (ii) $|I'| \leq f(k)$ with $f: \mathbb{N} \rightarrow \mathbb{N}$.

Typically, kernelizations are based on *data reduction rules* executable in polynomial time that help shrinking the instance size. It is known that L has a problem kernel if and only if L is in FPT [35]. Further, if the function f in the above kernelization definition is polynomial, then we also say that L has a *polynomial-size problem kernel*.

Parameterized intractability. As a central tool for classifying problems, Parameterized Complexity Theory provides the *W-hierarchy* consisting of the following classes and interrelations [37, 49]:

$$\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \dots \subseteq \text{W}[t] \subseteq \dots \subseteq \text{XP}.$$

To show $W[t]$ -hardness for any positive integer t , we use the concept of *parameterized reduction*. Let L, L' be two parameterized problems. A *parameterized reduction* from L to L' consists of two functions $f: \Sigma^* \rightarrow \Sigma^*$ and $g: \mathbb{N} \rightarrow \mathbb{N}$ such that for any given instance (I_L, k) of L , it holds that

- (i) (I_L, k) is a yes-instance for L if and only if $(f(I_L), g(k))$ is a yes-instance for L' , and
- (ii) f is computable in FPT time for parameter k , that is, in $h(k) \cdot |I_L|^{O(1)}$ time.

Problem L' is $W[t]$ -hard if for any problem L in $W[t]$, there is a parameterized reduction from L to L' . Typically, to show that some problem L' is $W[t]$ -hard, we start from some known $W[t]$ -hard problem L and reduce L to L' .

3 Nine Challenges

In this main section of our paper, we describe nine challenges (indeed, mostly rather problem areas) that found our specific interest and that shall help to stimulate fruitful research on the parameterized complexity of computational social choice problems.

3.1 ILP-Based Fixed-Parameter Tractability

Our first challenge relates to a method of establishing fixed-parameter tractability results. Let us explain this with the help of one of the most obvious parameters in the context of voting—the number of alternatives. For many contexts (for example, political or committee voting) it is natural to assume that the number of alternatives is small (particularly when compared to the number of voters). Hence, it is important to determine the computational complexity of various voting problems for the case where the input contains only few alternatives. Indeed, there is a number of fixed-parameter tractability results in terms of the parameter “number m of alternatives” in the voting context, but many of them rely on a deep result from combinatorial optimization due to Lenstra [78]. Moreover, since this result (on Integer Linear Programming) is mainly of theoretical interest, this may render corresponding fixed-parameter tractability results to be of classification nature only. Fixed-parameter tractability results based on Integer Linear Programming also tend to give less insight into the structural properties of the problems than combinatorial algorithms. The challenge we pose relates to improving this situation by replacing integer linear programs with direct combinatorial fixed-parameter algorithms.

Integer Linear Programming (ILP) is a strong classification tool for showing fixed-parameter tractability [87]. More specifically, Lenstra’s famous result [78] (see the literature [66, 74] for some moderate later running time improvements) implies that a problem is fixed-parameter tractable if it can be solved by an integer linear program where the number of variables is upper-bounded by a function solely depending on the considered parameter.

Perhaps the first example for such an “ILP-based” fixed-parameter tractability result in the context of computational social choice was implicitly given by Bartholdi III et al. [9] and later improved by McCabe-Dansted [86]. They developed an integer linear program to solve the NP-hard voting problem DODGSON SCORE and gave a running time bound based on Lenstra’s result. Without having explicitly stated this in their publication, this result yields fixed-parameter tractability for DODGSON SCORE with respect to the parameter number m of alternatives.

Before coming to their integer linear program formulation for DODGSON SCORE, we start with a brief definition of Dodgson voting. The input of DODGSON SCORE is an election $E = (C, V)$, a distinguished alternative $c \in C$, and an integer k . The question is whether one can make c the Condorcet winner by swapping a total number of at most k pairs of neighboring alternatives in the voters' preference orders. Refer to the literature [15, 9, 73] for more on the computational complexity status of DODGSON SCORE.

Bartholdi III et al. [9]'s integer linear program for DODGSON SCORE reads as follows. It computes the Dodgson score of alternative c , which is the minimum number of swaps of neighboring pairs of alternatives in order to make c the Condorcet winner.

$$\begin{aligned} \min \sum_{i,j} j \cdot x_{i,j} \text{ subject to} \\ \forall i \in \tilde{V} : \sum_j x_{i,j} = N_i \\ \forall y \in C : \sum_{i,j} e_{i,j,y} \cdot x_{i,j} \geq d_y \\ x_{i,j} \geq 0 \end{aligned}$$

In the above integer linear program, \tilde{V} denotes the set of preference order types (that is, the set of *different* preference orders in the given election), N_i denotes the number of voters of type i , $x_{i,j}$ denotes the number of voters with preference order of type i for which alternative c will be moved upwards by j positions, $e_{i,j,y}$ is 1 if the result of moving alternative c by j positions upward in a preference order of type i is that c gains an additional voter support against alternative y , and 0 otherwise. Furthermore, d_y is the deficit of c with respect to alternative y , that is, the minimum number of voter supports that c must gain against y to defeat y in a pairwise comparison. If c already defeats y , then $d_y = 0$. See Bartholdi III et al. [9] for more details. Altogether, the integer linear program contains at most $m \cdot m!$ variables $x_{i,j}$, where m denotes the number of alternatives. Thus, the number of variables in the described integer linear program is upper-bounded by a function in parameter m yielding fixed-parameter tractability due to Lenstra's result.

Although solvability by an integer linear program with a bounded number of variables implies fixed-parameter tractability, there is by far no guarantee for practically efficient algorithms. Indeed, due to a huge exponential function in the number of variables being part of the running time bound, the above fixed-parameter tractability result is basically for classification.

There are numerous further results [2, 16, 29, 17, 46, 60] for voting problems with the parameter number of alternatives where so far only using Lenstra's result leads to fixed-parameter tractability. In summary, this gives the following research challenge.

Key question 1: Can the mentioned ILP-based fixed-parameter tractability results be replaced by direct combinatorial (avoiding ILPs) fixed-parameter algorithms?

Interestingly, the KEMENY SCORE problem (refer to Section 3.5 for the definition) is known to be solvable in $O(2^m \cdot m^2 \cdot n)$ time due to a dynamic programming algorithm, that is, there is a direct combinatorial fixed-parameter algorithm [13]. Nonetheless, in practical instances using an Integer Linear Programming formulation with $O(m^2)$ binary variables is much more efficient, although its theoretical running time is significantly worse [12].

3.2 Parameterized Complexity of Bribery Problems

Our second challenge regards the parameterized complexity of election bribery problems under some voting rule \mathcal{R} . There are several families of bribery problems but, generally speaking, their main idea is as follows. We are given some election $E = (C, V)$, some preferred alternative p , and some budget B . If we choose to bribe some voter v , then we can modify v 's preference order, but we have to pay for it (in some variants of the bribery problem the cost depends on how we change the preference order, whereas in others the cost is fixed irrespective of the extent to which we modify the preference order). The goal is to compute which voters to bribe—and how to modify their preference orders—so that p becomes an \mathcal{R} -winner and the bribery cost is at most B .

The study of the computational complexity of election bribery was initiated by Faliszewski et al. [57] (also refer to the work of Faliszewski and Rothe [62] for a survey). In particular, they studied the following two problems for some voting rule \mathcal{R} :

- In the \mathcal{R} -BRIBERY problem, each voter has the same price (unit cost) for being bribed. In effect, we ask if it is possible to ensure the preferred alternative's victory by bribing at most B voters.
- In the \mathcal{R} -\$BRIBERY problem, each voter v has an individual price π_v for being bribed.

Note that in both problems the cost of bribing a voter does not depend on how the voter's preference order is changed: The briber pays for the ability to change the preference order in any convenient way.

Later, Elkind et al. [54] introduced another variant of the bribery problem, which they called \mathcal{R} -SWAP BRIBERY, where the cost of bribing a voter depends on the extent to which we modify this voter's preference order. Formally, they required that each voter v has a swap-bribery price function, which for each two alternatives a and b gives the cost of convincing v to swap a and b in the preference order (provided that a and b are adjacent in this preference order at the time). To bribe a voter v , we provide a sequence of pairs of alternatives to swap, the voter looks at these pairs one by one and for each pair swaps the alternatives in the preference order (provided they are adjacent at the time) and charges us according to the swap-bribery function. Elkind et al. [54] also defined a simpler variant of SWAP BRIBERY, called SHIFT BRIBERY, where all the swaps have to involve the preferred alternative. While both SWAP BRIBERY and SHIFT BRIBERY tend to be NP-complete for typical voting rules, SHIFT BRIBERY is much easier to solve approximately [52].

For most typical voting rules, these bribery problems are NP-complete. A few polynomial-time algorithms exist only for the simplest of voting rules, such as Plurality, or d -Approval and Bucklin, for the case of SHIFT BRIBERY. However, from the point of view of parameterized complexity theory, there is a huge difference between the \mathcal{R} -BRIBERY problems, where each voter has unit cost for being bribed, and the other flavors of bribery, where each voter has individually specified price. Indeed, when we use the number of alternatives as the parameter, for typical voting rules \mathcal{R} we have that \mathcal{R} -BRIBERY is in FPT (this is implicit in the work of Faliszewski et al. [57]), whereas the other types of bribery are in XP [57, 54] and it is not known if they are in FPT or hard for W[1], W[2], or some further class in the W-hierarchy. Thus, we have the following challenge.

Key question 2: For each typical voting rule \mathcal{R} : what is the exact parameterized complexity of problems \mathcal{R} - $\$$ BIBERY, \mathcal{R} -SWAP BIBERY, and \mathcal{R} -SHIFT BIBERY when parameterized by the number of alternatives?

Interestingly, this challenge is very closely related to the previous one. For many typical voting rules \mathcal{R} , the proof that \mathcal{R} -BIBERY is in FPT uses an ILP-based algorithm. Having such an algorithm (instead of a combinatorial one) gives very limited insight into the nature of the problem and, thus, we cannot use the ideas regarding \mathcal{R} -BIBERY to build algorithms for, say, \mathcal{R} - $\$$ BIBERY. On the contrary, there is no obvious way to extend the ILP-based algorithms to the variants of bribery where voters have individual prices. The general idea behind these ILP algorithms is to consider groups of voters with the same preference orders jointly, but this does not lead to correct solutions when these voters have differing prices.

Naturally, some authors have already studied parameterized complexity of various types of bribery problems [46, 30]. However, their results regard either parameters other than the number of alternatives, or special pricing schemes where voters can be treated as unified groups and ILP approaches work.

3.3 FPT Approximation Algorithms

In the previous challenge we asked about the exact complexity of various election bribery problems parameterized by the number of alternatives. However, instead of seeking exact complexity results it might be easier to find good FPT approximation algorithms (refer to the work of Marx [85] for a survey). After all, if the bribery problems turn out to be, say, W[1]- or W[2]-hard, the approximation algorithms would give us some means of solving them. Even if the problems turned out to be in FPT, it is likely that the approximation algorithms would be much faster in practice. Further, even though for some of these problems there are good polynomial-time approximation algorithms (for example, this is the case for SHIFT BIBERY under positional scoring rules [52]), it might be the case that FPT approximation algorithms would yield better approximation ratios.

To see that FPT approximation algorithms can indeed outperform polynomial-time ones in terms of achievable approximation ratios, let us consider the MAX VERTEX COVER problem (also known as PARTIAL VERTEX COVER). In this problem we are given an undirected graph G and an integer k . The goal is to pick k vertices that jointly *cover* as many edges as possible. Note that a vertex *covers* all its incident edges. As a generalization of VERTEX COVER, the problem is NP-complete [67], and it is also known to be W[1]-complete when parameterized by k [71]. As far as we know, the best polynomial-time approximation algorithm for this problem known at the moment is due to Ageev and Sviridenko [1]. This algorithm achieves approximation ratio $\frac{3}{4}$, that is, if the optimal solution covers OPT edges, the algorithm of Ageev and Sviridenko guarantees to cover at least $\frac{3}{4}\text{OPT}$ edges. Marx [85], however, has shown an algorithm that for each positive number ε finds a solution for MAX VERTEX COVER that covers at least $(1 - \varepsilon)\text{OPT}$ edges and runs in FPT time with respect to parameters k and ε . In other words, Marx has given an FPT approximation scheme for the problem. Recently, this result of Marx was generalized by Skowron and Faliszewski [94]. Interestingly, the motivation for their work came from a computational social choice context.

This discussion motivates the following research challenge:

Key question 3: For which computationally hard election-related problems are there FPT approximation schemes?

As we have indicated, bribery problems are perhaps the most natural ones to consider here. Indeed, Bredereck et al. [30] gave a fairly general FPT approximation scheme for SHIFT BRIBERY parameterized by the number of voters, and an FPT approximation scheme for SHIFT BRIBERY parameterized by the number of alternatives, but for a somewhat restrictive model of bribery costs. The existence of an FPT approximation scheme that does not rely on assumptions about the pricing model remains open. Schlotter et al. [92] provided an FPT approximation scheme for yet another bribery problem, called SUPPORT BRIBERY. Interestingly, we are not aware of any FPT approximation algorithms for the BRIBERY and \$BRIBERY families of problems.

There are also many other types of election problems one could try to seek FPT approximation algorithms for. For example, in control problems we are given an election $E = (C, V)$, a preferred alternative p , and some means of affecting the structure of the election. For example, we might have some set of additional alternatives or voters that can be convinced to participate in the election, or we might have some means to remove some of the alternatives or voters. The task is to ensure that p is a winner while modifying the election as little as possible. Election control problems were introduced by Bartholdi III et al. [8] (later Hemaspaandra et al. [72] extended the definition to include so-called destructive control) and were studied from the parameterized complexity perspective by several authors [18, 81, 82, 36]. However, at the intuitive level, it seems that for control problems it might be easier to obtain parameterized inapproximability results. The reason for that is that often their hardness proofs (parameterized or not) directly rely on problems that are difficult to approximate.

3.4 Kernelization Complexity of Voting Problems

Kernelization is one of the major tools to prove fixed-parameter tractability results. In the last decade, parameterized complexity theory witnessed an extremely rapid development of kernelization results. Many problems turn out to admit polynomial-size problem kernels, most of them being graph-theoretical problems [70, 22, 84].

With respect to voting problems, although many have been shown to be fixed-parameter tractable, only very few problems are known to admit polynomial-size problem kernels. The main reason for this might be that a voting problem usually contains as input an election whose size essentially is upper-bounded by the number m of alternatives times the number n of voters. According to the definition of polynomial-size problem kernels, we need to bound both numbers by polynomial functions in the considered parameter; it already seems difficult to bound one out of n and m with a function of the other. Next, we use the CONSTRUCTIVE CONTROL BY DELETING VOTERS for d -Approval (CCDV- d -APPROVAL) problem to illustrate this difficulty and discuss how to cope with it.

Given an election (C, V) under the d -Approval rule, each voter assigns one point to each of the alternatives c ranked in the top d positions of his preference order. We say then that this voter approves c . The score $s(c)$ of an alternative c is the total number of points it gains from all the voters. The alternative with the highest score wins the election. The CCDV- d -APPROVAL problem asks for a list of at most k voters whose deletion from V makes a distinguished alternative p win the election. CCDV- d -APPROVAL is NP-hard for every constant $d \geq 3$ [80].

It is not hard to show that CCDV- d -APPROVAL is fixed-parameter tractable with respect to the number k of deleted voters. Observe that the alternatives c with $s(c) < s(p)$ are *irrelevant* for control by deleting voters since the voters approving p will never be deleted from V . Let I denote the set of all irrelevant alternatives, and $R := C \setminus (I \cup \{p\})$. Since each alternative c in R satisfies $s(c) \geq s(p)$, we have to delete at least one voter from V who approves c . By the definition of d -Approval, deleting k voters can decrease the scores of at most $d \cdot k$ alternatives, and thus $|R| \leq d \cdot k$. Let V_p be the list of voters in V approving p and V_R the list of voters approving at least one alternative in R but not p . Obviously, the voters deleted should be all from V_R . With $|R| \leq d \cdot k$, we can partition V_R into at most $O((d \cdot k)^d)$ many classes, each class containing the voters in V_R which approve the same subset of R . A brute-force algorithm can then be applied to guess the number of voters to be deleted from each class, resulting in an FPT running time.

With respect to deriving a polynomial problem kernel for CCDV- d -APPROVAL, we further need to bound $|I|$ and $|V|$. Deleting voters who solely approve the alternatives in I do not help making p win and can be safely “removed” from the voter list V . Thus, we can assume that the voters in V who approve an irrelevant alternative should also approve p or at least one alternative from R . This means that $V = V_p \cup V_R$. Therefore, the key is to bound $|V_p|$ and $|V_R|$. However, although the number of the classes of voters in V_R is bounded, the number of voters in each class can be unbounded. The same holds also for V_p , even if none of the voters in V_p will be deleted. Due to the relation between the scores of p and of the alternatives in R , we cannot reduce V_p and V_R independently. The kernelization algorithm of Wang et al. [96] performs a “reconstruction” approach, that is, it keeps only the “essential” part of the instance, which consists of particular voters in V_R , and then constructs a new instance around this essential part with new irrelevant alternatives and some new voters. The decisive idea behind this approach is to restore the score difference between $s(p)$ and $s(c)$ for every $c \in R$ after the reconstruction; but $s(p)$ and $s(c)$ are much less than in the original instance and, thus, the number of voters needed can be bounded by a function of k .

The kernelization algorithm of Wang et al. [96] achieves a polynomial problem kernel for CCDV- d -APPROVAL with d being a constant; note that, with unbounded d , CCDV- d -APPROVAL is W[2]-hard with respect to k [96]. It is conceivable that this reconstruction approach could lead to problem kernels for other fixed-parameter tractable control problems [83]. However, compared to the diverse general tools for deriving polynomial (even linear) problem kernels for graph-based problems, the research on kernelization for such two-dimensional problems as voting problems seems little developed. Only very few problem kernel results for voting problems are known so far. Thus, to close this gap is a major challenge.

Key question 4: What is the kernelization complexity of fixed-parameter tractable voting problems with respect to the number m of alternatives, the number n of voters, or some parameter less than m or n ? Can we derive polynomial (or even linear) problem kernels for some voting problems with the above parameters?

3.5 Partial Problem Kernels

As already discussed in the previous section, the input to a voting problem contains an election whose size is upper-bounded by the number of alternatives times the number of voters. In addition, the input can also contain prices per voter or per position for each voter’s preference

$$\begin{aligned}
v_1: a_1 &\succ a_2 \succ a_3 \succ a_4 \\
v_2: a_1 &\succ a_2 \succ a_4 \succ a_3 \\
v_3: a_3 &\succ a_2 \succ a_1 \succ a_4
\end{aligned}$$

Figure 1: An election with four alternatives and three voters. The unique Kemeny ranking is $a_1 \succ a_2 \succ a_3 \succ a_4$ with score four.

order. Either the number of alternatives, the number of voters, or the total price are promising as a parameter for performing a parameterized complexity analysis because there are applications where either of these parameters is relatively small when compared to the other. For instance, in aggregating the outcomes of different search engines with the help of a meta-search engine, the number of voters is (relatively) small but the number of alternatives is large. In political voting, the situation should typically be the other way round. In terms of developing kernelization results, however, the current knowledge indicates a certain asymmetry with respect to both parameters, “the number m of alternatives” and “the number n of voters”. While we have often positive results when exploiting parameter m , for parameter n fewer positive results are known [11]. Indeed, this also motivates the concept of “partial problem kernels”.

Formally, a decision problem L with input instance (I, k) is said to admit a *partial kernel* if there is a computable function $d: \Sigma^* \rightarrow \mathbb{N}$ such that

- (i) L is decidable in FPT time for parameter $d(I)$, and
- (ii) there is a polynomial-time algorithm that given (I, k) , computes an equivalent instance (I', k') such that $k' \leq f(k)$ and $d(I') \leq g(k)$ with $f, g: \mathbb{N} \rightarrow \mathbb{N}$.

We will discuss this issue using the rank aggregation problem KEMENY SCORE as a concrete example.

Informally speaking, KEMENY SCORE asks, given an election $E = (C, V)$ and a positive number k , whether there is a median *ranking*, that is, a linear order over C whose total number of inversions with the voters’ preference orders from E is at most k . The score of such a ranking is the total number of inversions needed. A *Kemeny ranking* of an election is a median ranking with minimum score. An example election with four alternatives and three voters is illustrated in Figure 1. It has only one Kemeny ranking $a_1 \succ a_2 \succ a_3 \succ a_4$ with score four.

Surprisingly, KEMENY SCORE remains NP-hard when the input election E has only four voters [50, 51, 19]. In particular, this implies that there is no hope for fixed-parameter tractability with respect to the parameter “number of voters”. On the contrary, it is straightforward to observe that a simple brute-force search already yields fixed-parameter tractability with respect to the number of alternatives [13].

As a matter of fact, there are further natural parameterizations of KEMENY SCORE [13]. In particular, one may ask whether KEMENY SCORE becomes (more) tractable if the preference orders of the input election are very similar on average, that is, the sum over their pairwise differences divided by the number of pairs is small. Let us call the smallest integer at least as large as this value parameter d_a . Developing a dynamic programming algorithm [13], it has been shown that KEMENY SCORE is fixed-parameter tractable for parameter d_a with running

time $16^{d_a} \cdot \text{poly}(n + m)$ where poly is a polynomial function. Based on this, one immediately obtains a trivial problem kernel of size $O(16^{d_a})$ for parameter d_a . A natural follow-up question concerns the existence of a problem kernel of smaller size or even of polynomial size for KEMENY SCORE when parameterized by d_a . We only have partially positive results here. The question whether we can upper-bound both the number of alternatives and the number of voters by a polynomial function in d_a by means of polynomial-time data reduction remains open. It is, however, possible to upper-bound the number of alternatives by a linear function in d_a [12, 93]. Since KEMENY SCORE is decidable in FPT time for m , it admits a partial kernel. This concept has also been used in other application contexts, some of them outside the field of computational social choice [14, 10, 30].

Partial problem kernels of polynomial size have been shown to be useful to improve fixed-parameter tractability results. It is unclear, however, whether or not some of the partial kernelization results mentioned can be replaced by “full kernelization” results. This leads to the following challenge.

Key question 5: Can the known partial problem kernels be improved to (full) problem kernels with non-trivial size bounds?

We mention in passing that, focusing on voting problems, it might help to use weights for voters to obtain (full) problem kernels with non-trivial size bounds; however, this would not be a “plain” kernelization (where the kernelized instance needs to be of “same type” as the input instance) in case of unweighted input voters and may hide computational complexity in the weight functions by modifying the problem.

3.6 Parameterizations of Election Data

Social choice theory is full of (combinatorial and algorithmic) results on voting problems. Many of these results are centered around *general* elections, where *every* ordering of the alternatives is a feasible preference for any voter and where *every* combination of alternative orderings yields a feasible election. For instance, Arrow’s impossibility theorem shows the impossibility of preference aggregation for *general* elections under certain axioms (unrestricted domain, non-dictatorship, Pareto efficiency, and independence of irrelevant alternatives). Another example is the result of Hemaspaandra et al. [73] that establishes hardness of determining the winner in Dodgson elections with *general* elections.

For this reason, one research branch concentrates on the investigation of specially structured elections, where only certain combinations of alternative orderings are feasible.

- An election is *single-peaked* if there exists a linear order of the alternatives such that any voter’s preference along this order is either (i) always strictly increasing, (ii) always strictly decreasing, or (iii) first strictly increasing and then strictly decreasing. The example in Figure 2 shows a single-peaked election with five alternatives and three voters.
- An election is *single-crossing* if there exists a linear order of the voters such that for any pair of alternatives along this order, either (i) all voters have the same opinion on the ordering of these two alternatives or (ii) there is a single spot where the voters switch from preferring one alternative to the other one.

$$\begin{aligned}
v_1: a_1 \succ a_2 \succ a_3 \succ a_4 \succ a_5 \\
v_2: a_3 \succ a_4 \succ a_2 \succ a_1 \succ a_5 \\
v_3: a_3 \succ a_2 \succ a_1 \succ a_4 \succ a_5
\end{aligned}$$

Figure 2: An election with five alternatives and three voters. It is single-peaked with respect to the orders $a_1 \succ a_2 \succ a_3 \succ a_4 \succ a_5$ and $a_4 \succ a_3 \succ a_2 \succ a_1 \succ a_5$, and to their reverse orders.

- An election is *one-dimensional Euclidean* if there exists an embedding of alternatives and voters into the real numbers, such that every voter prefers the nearer one of each pair of alternatives.

It is known that Arrow’s impossibility result disappears on single-peaked elections [20], single-crossing elections [90], and on one-dimensional Euclidean elections (which form a special case of single-peaked and of single-crossing elections; see also the paper of Elkind, Faliszewski, and Skowron [53] for some recent discussion of elections that are both single-peaked and single-crossing). On the algorithmic side, Walsh [95], Brandt et al. [27], and Faliszewski et al. [61] showed that many electoral bribery, control, and manipulation problems that are NP-hard in the general case become tractable under single-peaked elections.

The three restrictions allow natural parameterizations. Yang and Guo [97] considered *k-peaked* elections as a generalization of single-peaked elections, every preference order can have at most k peaks (that is, at most k rising streaks that alternate with falling streaks). Similarly, we can generalize single-crossing elections to *k-crossing* elections, where for every pair of alternatives there are at most k spots where the voters switch from preferring one alternative to the other one. A natural generalization of one-dimensional Euclidean elections are *k-dimensional Euclidean* elections, where alternatives and voters are embedded in k -dimensional Euclidean space.

Key question 6: How does the complexity of the standard bribery, manipulation and control problems in election systems depend on the parameter k for (i) *k-peaked*, (ii) *k-crossing*, (iii) *k-dimensional Euclidean* elections?

A first step in this direction would be to get a good combinatorial understanding of such elections. The literature contains forbidden substructure characterizations for single-peaked elections [7] and for single-crossing elections [32]. Knoblach [76] discussed the structure of one-dimensional Euclidean elections. Chen et al. [38] studied minimal the relation between single-peaked and single-crossing elections and one-dimensional Euclidean elections. Bogomolnaia and Laslier [24], and Bulteau and Chen [33] investigated the restrictiveness of *k-dimensional Euclidean* elections. To our knowledge, the combinatorial structure of the above parameterized variants has not been investigated so far.

3.7 Distance to Tractable Cases

“Treewidth” is a famous concept in algorithmic graph theory. Informally speaking, it uses a natural number to express how far a given undirected graph is away from being a tree: a graph with treewidth one is a tree or a forest, and the smaller the treewidth is, the closer the graph is

to being a tree [21, 23, 44]. The interest for treewidth stems from the fact that many NP-hard graph problems can be solved efficiently (say by greedy or dynamic programming algorithms) when restricted to trees. Hence, it is natural to ask whether such tractability results can be extended beyond trees. Treewidth is one measure that turned out particularly useful for such a quest. More specifically, many NP-hard graph problems (including CLIQUE, INDEPENDENT SET, DOMINATING SET) can be solved in linear time on graphs with bounded treewidth (assuming that a corresponding “tree decomposition” is part of the input) [21, 23, 48, 87]. Put differently, these problems are fixed-parameter tractable with respect to the parameter “treewidth” of the input graph.

Also in the context of voting, there exist certain structural properties of inputs that make some of the otherwise NP-hard problems tractable. For instance, under single-peakedness (as discussed in Section 3.6) many computationally hard winner determination problems, including KEMENY SCORE [27] (as discussed in Section 3.5), turn out to be polynomial-time solvable. In analogy to the step from actual trees to “nearly” trees (that is, graphs with bounded treewidth), the computational social choice community has taken the step from actual single-peakedness to “nearly” single-peakedness, all in the spirit of pushing the borders of tractability. The following notions of nearly single-peaked elections have been studied in [56, 42, 31, 59, 41].

- Elections that can be made single-peaked by deletion of k voters (also called *Maverick* voters).
- Elections that can be made single-peaked by deletion of k alternatives.
- Elections that can be made single-peaked by k swaps in the preferences of every voter.
- Elections that can be made single-peaked by contracting groups of up to k alternatives into single alternatives (here the alternatives of every contracted group must show up consecutively in the preferences of every voter).

Each of these notions introduces a distance measure (or *width* measure) to the tractable case of single-peakedness. In the same spirit, Elkind et al. [55], Bredereck et al. [31], and Cornaz et al. [42] introduced distance measures of elections to the tractable case of single-crossingness (as discussed in Section 3.6) and to the tractable case of group separability (an election is group separable if the alternatives can be partitioned into two non-empty groups such that every voter prefers all alternatives of one group to all alternatives of the other group).

Altogether, these approaches lead to natural and meaningful “width-bounded” measures for preference orders which shall, in the same spirit as treewidth and other width-related parameters do for graphs, lead to interesting special cases allowing to obtain fixed-parameter tractability results for social choice problems with respect to these width parameters. Of course, coming up with natural and meaningful “width-bounded” measures is an ongoing challenging task. Nonetheless, considering the huge amount and great success of work on width measures for graphs, there appear to be many research opportunities for working on these measures in the context of voting and related problems. We remark that, from the viewpoint of parameterized complexity analysis, such investigations would fall under the category “distance from triviality”-parameterizations [34, 69]. For applications in computational social choice, this line of research is still in its infancy with few results [42], leading to the following generic challenge.

Key question 7: How can natural and meaningful “width-based” parameters be used to gain (practically useful) fixed-parameter tractability results for NP-hard voting problems?

3.8 W-Hierarchy and Majority-Based Problems

This theoretical challenge is about creating a framework that could simplify the W-hierarchy classification of problems using majority-based properties which naturally occur in Computational Social Choice.

Problems that are presumably not fixed-parameter tractable are usually classified in the W-hierarchy which is defined via the WEIGHTED WEFT- t DEPTH- d CIRCUIT SATISFIABILITY (WCS(t, d)) problem [48].

The input of WCS(t, d) is a Boolean circuit (formally, these are vertex-labeled directed acyclic graphs) of *depth* d and *weft* t , and an integer bound k . The question is whether there is a satisfying assignment of weight k . In this context, Boolean circuits are allowed to contain NOT gates, small AND and OR gates of fan-in at most two and large AND and OR gates of arbitrary finite fan-in. The weft of a Boolean circuit is the maximum number of large gates on a path from an input to the output. The depth is the maximum number of all gates on a path from an input to the output. The weight of an assignment is the number of variables set to TRUE.

For any $t \geq 1$, $W[t]$ is the class of parameterized problems for which a parameterized reduction to WCS(t, d) exists for some constant $d \geq 1$ (regarding the parameter k).

Classifying a parameterized problem in the W-hierarchy consists of two parts. The first part is to show $W[t]$ -membership for some integer t . This is often done by providing a parameterized reduction to WCS(t, d) for some d or to a parameterized problem which is already known to be in $W[t]$.

The second part is to find a lower bound on the complexity, that is, to find some integer t' such that the problem is $W[t']$ -hard. This is usually done by describing a parameterized reduction from some known $W[t']$ -hard problem. In the ideal case, where $t = t'$, one precisely classifies the problem within the W-hierarchy. It is also possible that one cannot find a t such that the problem is in $W[t]$, because the problem is $W[t']$ -hard for all t' or $W[P]$ -hard or LOGSNP-hard [89] or NP-hard for constant parameter values.

Sometimes showing $W[t]$ -membership appears to be more challenging than showing hardness. This is due to the nature of the investigated problem. For instance, a problem which is based on some majority properties cannot be easily formulated via Boolean circuits containing only NOT, AND, and OR gates, because the majority operator cannot be easily simulated via a single large AND or OR gate.

A concrete example is the MAJORITYWISE ACCEPTED BALLOT (MAJAB) problem. Its input is a set \mathcal{P} of m proposals, a society of n voters with favorite ballots $B_1, \dots, B_n \subseteq \mathcal{P}$, and an agenda $Q_+ \subseteq \mathcal{P}$. The question is whether there is a ballot Q , $Q_+ \subseteq Q \subseteq \mathcal{P}$, such that a *strict majority* of the voters *accepts* Q . A voter i accepts ballot Q if $|B_i \cap Q| > |Q|/2$. MAJAB is $W[2]$ -hard for the parameter size $|Q|$ of the accepted ballot even if $Q_+ = \emptyset$ [2]. However, showing $W[2]$ -membership seems challenging. Note that the *unanimous* version of this problem, where one asks for a ballot $Q \subseteq \mathcal{P}$ that is accepted by all voters, is $W[2]$ -complete.

Modifying the set of allowed gates in WCS(t, d) dramatically simplifies the task of finding a membership proof but it may lead to a slightly different W-hierarchy: For instance, we can

show that MAJAB can be reduced to a $WCS(2, d)$ problem variant, where instead of NOT, AND, or OR gates, majority gates (which output TRUE if the majority of inputs is TRUE) are used. This version of the problem which we call $WCS(t, d)(MAJ)$, and the corresponding $W(Maj)$ -hierarchy have been studied by Fellows et al. [63]. While $W[1] = W[1](MAJ)$, for $t \geq 2$, it is only known that $W[t] \subseteq W[t](MAJ)$. It would be interesting to know whether $W[t](MAJ) \subseteq W[t]$ also holds. If one cannot show this, then it would be nice to have $W[t](MAJ)$ -complete problems to show intractability results.

Key question 8: How does the $W(Maj)$ -hierarchy relate to the classical W -hierarchy? What are accessible complete problems for the $W(Maj)$ -hierarchy?

Besides majority variants of SET COVER and HITTING SET (see [63, Sec. 7]) promising natural candidates like MAJAB for $W[2](MAJ)$ -complete problems may occur in the context of computational social choice, where majority-based properties such as Condorcet winner frequently occur.

3.9 Cake Cutting

In the cake division problem, $n \geq 2$ players want to cut a cake into n pieces so that every player gets a “fair” share of the cake according to his own private measure. The cake represents a heterogeneous divisible good, and usually is modeled as the unit-interval $\mathcal{C} = [0, 1]$. Every player p ($1 \leq p \leq n$) has his own measure μ_p on the subsets of \mathcal{C} . These measures are assumed to be well-behaved, and to satisfy the following conditions:

- Positivity: $\mu_p(X) \geq 0$ for all $\emptyset \neq X \subseteq \mathcal{C}$.
- Additivity: $\mu_p(X) + \mu_p(X') = \mu_p(X \cup X')$ for disjoint subsets $X, X' \subseteq \mathcal{C}$.
- Continuity: For every $X \subseteq \mathcal{C}$ and for every λ with $0 \leq \lambda \leq 1$, there exists a subset $X' \subseteq X$ such that $\mu_p(X') = \lambda \cdot \mu_p(X)$.
- Normalization: $\mu_p(\mathcal{C}) = 1$.

A cake division is *proportional* if every player receives a piece that he values at least $1/n$; the division is *envy-free* if every player p gets a piece that he values at least as much as the piece of any other player; it is *equitable* if every player p gets a piece that he values exactly $1/n$. A *protocol* is a procedure that controls the division process of the cake \mathcal{C} .

The book [91] by Robertson and Webb provides an excellent summary of the area, and in particular covers all kinds of algorithmic aspects. There already exists a rich literature on cake cutting, and various authors have come up with dozens of proportional or envy-free protocols. In most of the known protocols, the measures μ_p enter the game as a black box and can be queried and evaluated for given cake pieces; the standard goal then is to design protocols that minimize the total number of queries. For getting algorithmic problems in our classical algorithmic sense, however, we have to fully specify the problem data; in particular, the measures μ_p must be given explicitly (and not just as a black box!) and must be represented succinctly. A natural approach is to represent the measures through piecewise linear value density functions $f_p : [0, 1] \rightarrow \mathbb{R}$. A feasible piece of cake then consists of finitely many intervals, and the corresponding measure is

the integral of f_p taken over all these intervals; note that the normalization condition implies that the integral of f_p over the full interval $[0, 1]$ equals 1.

Brams et al. [26] provided a three-player example with piecewise linear value density functions that has (i) an envy-free division and (ii) an equitable division, but that does not allow a division that is simultaneously envy-free and equitable. Kurokawa et al. [77] also considered value density functions that are piecewise linear. They design an envy-free cake cutting protocol which requires a number of operations that is polynomial in some natural parameters of the given instance. Aumann et al. [6] considered value density functions that are piecewise constant, and additionally require that each player ends up with a *single* subinterval of the cake $[0, 1]$; in this model it is NP-hard to find a division that maximizes the utilitarian welfare, that is, the sum of the values that the n players assign to their respective pieces.

There are some straightforward parameterizations that naturally generalize the models that we just discussed: The value density function of every player consists of at most α pieces, and each such function piece is a polynomial function of degree at most β . A feasible division of the unit-interval cake (for n players) consists of at most γn subintervals altogether, and every player receives at most δ subintervals. Instead of using the unit-interval, the cake could also be modeled as the 2-dimensional unit-square, or as some simple d -dimensional object (with the dimension d as parameter); the subintervals then would translate into simple polyhedral pieces with a small number of facets (and with a number of additional parameters for measuring simplicity).

Key question 9: How hard is it to find proportional or envy-free or equitable divisions under the above parameterizations?

Clearly, there is an abundance of other ways for parameterizing the cake-cutting world. As yet another challenging objective function, we mention the egalitarian welfare, the minimum of the values that the n players assign to their respective pieces.

4 Conclusion

The purpose of this work is to stimulate more research in a promising application field of computational complexity analysis and algorithmics. Numerous further research challenges can be found in the very active and steadily growing area of computational social choice. Clearly, our selection of topics was subjective and could be easily extended. For instance, in voting contexts it often makes sense (motivated by real-world data) to deal with partial instead of linear orders, generally making the considered problems harder. Here, one might parameterize on the “degree of non-linearity”. Additional interesting (broad) research topics would be to further explore on relations to fields such as scheduling or graph and matching theory. By their very nature, many problems in computational social choice carry many natural parameterizations, motivating a thorough multivariate computational complexity analysis [64, 88]. Finally, so far there is little work in terms of algorithm engineering for empirically validating and improving the performance of fixed-parameter algorithms for social choice problems.

Acknowledgments. We thank Britta Dorn, Dominikus Krüger, Jörg Rothe, Lena Schend, Arkardii Slinko, Nimrod Talmon, and an anonymous referee for their constructive feedback on previous versions of the manuscript. Robert Brederick and Piotr Faliszewski were supported by

the Deutsche Forschungsgemeinschaft, project PAWS (NI 369/10). Jiehua Chen was supported by the Studienstiftung des Deutschen Volkes, Jiong Guo was supported by DFG “Cluster of Excellence Multimodal Computing and Interaction”, and Gerhard Woeginger was supported by DIAMANT (a mathematics cluster of the Netherlands Organization for Scientific Research NWO) and by the Alexander von Humboldt Foundation, Bonn, Germany.

References

- [1] A. A. Ageev and M. Sviridenko. Approximation algorithms for maximum coverage and max cut with given sizes of parts. In *Integer Programming and Combinatorial Optimization (IPCO’99)*, volume 1610 of *LNCS*, pages 17–30. Springer, 1999. → p. 7.
- [2] N. Alon, R. Brederbeck, J. Chen, S. Kratsch, R. Niedermeier, and G. J. Woeginger. How to put through your agenda in collective binary decisions. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT’13)*, volume 8176 of *LNCS*, pages 30–44, 2013. → pp. 5 and 14.
- [3] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. → p. 3.
- [4] K. J. Arrow, A. K. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare, Volume 1*. North-Holland, 2002. → p. 1.
- [5] K. J. Arrow, A. K. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare, Volume 2*. North-Holland, 2010. → p. 1.
- [6] Y. Aumann, Y. Dombb, and A. Hassidim. Computing socially-efficient cake divisions. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS’13)*, pages 343–350. IFAAMAS, 2013. → p. 16.
- [7] M. A. Ballester and G. Haeringer. A characterization of the single-peaked domain. *Social Choice and Welfare*, 36(2):305–322, 2011. → p. 12.
- [8] J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8/9):27–40, 1992. → p. 8.
- [9] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989. → pp. 4 and 5.
- [10] M. Basavaraju, M. C. Francis, M. S. Ramanujan, and S. Saurabh. Partially polynomial kernels for Set Cover and Test Cover. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS’13)*, volume 24 of *LIPIcs*, pages 67–78. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2013. → p. 11.
- [11] N. Betzler, R. Brederbeck, J. Chen, and R. Niedermeier. Studies in computational aspects of voting—a parameterized complexity perspective. In *The Multivariate Algorithmic Revolution and Beyond*, volume 7370 of *LNCS*, pages 318–363. Springer, 2012. → pp. 2 and 10.

- [12] N. Betzler, R. Bredereck, and R. Niedermeier. Theoretical and empirical evaluation of data reduction for exact Kemeny rank aggregation. *Autonomous Agents and Multi-Agent Systems*, 28(5):721–748, 2014. → pp. 5 and 11.
- [13] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond. Fixed-parameter algorithms for Kemeny rankings. *Theoretical Computer Science*, 410(45):4554–4570, 2009. → pp. 5 and 10.
- [14] N. Betzler, J. Guo, C. Komusiewicz, and R. Niedermeier. Average parameterization and partial kernelization for computing medians. *Journal of Computer and System Sciences*, 77(4):774–789, 2011. → p. 11.
- [15] N. Betzler, J. Guo, and R. Niedermeier. Parameterized computational complexity of Dodgson and Young elections. *Information and Computation*, 208(2):165–177, 2010. → p. 5.
- [16] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI’09)*, pages 53–58. AAAI Press, 2009. → p. 5.
- [17] N. Betzler, R. Niedermeier, and G. J. Woeginger. Unweighted coalitional manipulation under the Borda rule is NP-hard. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence (IJCAI’11)*, pages 55–60. AAAI Press, 2011. → p. 5.
- [18] N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theoretical Computer Science*, 410(52):43–53, 2009. → p. 8.
- [19] T. Biedl, F. J. Brandenburg, and X. Deng. On the complexity of crossings in permutations. *Discrete Mathematics*, 309(7):1813–1823, 2009. → p. 10.
- [20] D. Black. On the rationale of group decision making. *Journal of Political Economy*, 56(1):23–34, 1948. → p. 12.
- [21] H. L. Bodlaender. Treewidth: Characterizations, applications, and computations. In *Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG’06)*, volume 4271 of *LNCS*, pages 1–14, 2006. → p. 13.
- [22] H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *Proceedings of the 4th International Workshop on Parameterized and Exact Computation (IWPEC’09)*, volume 5917 of *LNCS*, pages 17–37. Springer, 2009. → pp. 3 and 8.
- [23] H. L. Bodlaender and A. M. C. A. Koster. Combinatorial optimization on graphs of bounded treewidth. *The Computer Journal*, 51(3):255–269, 2008. → p. 13.
- [24] A. Bogomolnaia and J.-F. Laslier. Euclidean preferences. *Journal of Mathematical Economics*, 43(2):87–98, 2007. → p. 12.
- [25] S. Brams and P. C. Fishburn. Voting procedures. In K. J. Arrow, A. K. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 1, pages 173–236. Elsevier, 2002. → p. 3.

- [26] S. J. Brams, M. A. Jones, and C. Klamler. N -person cake-cutting: There may be no perfect division. *The American Mathematical Monthly*, 120(1):35–47, 2013. → p. 16.
- [27] F. Brandt, M. Brill, E. Hemaspaandra, and L. A. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI’10)*, pages 715–722. AAAI Press, 2010. → pp. 12 and 13.
- [28] F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. In *Multiagent Systems*. MIT Press, 2012. → p. 1.
- [29] R. Bredereck, J. Chen, S. Hartung, S. Kratsch, R. Niedermeier, O. Suchý, and G. J. Woeginger. A multivariate complexity analysis of lobbying in multiple referenda. *Journal of Artificial Intelligence Research*, 50:409–446, 2014. → p. 5.
- [30] R. Bredereck, J. Chen, A. Nichterlein, P. Faliszewski, and R. Niedermeier. Prices matter for the parameterized complexity of shift bribery. In *Proceedings of the 28th Conference on Artificial Intelligence (AAAI’14)*, 2014. → pp. 7, 8, and 11.
- [31] R. Bredereck, J. Chen, and G. J. Woeginger. Are there any nicely structured preference profiles nearby? In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI’13)*, pages 62–68. AAAI Press, 2013. → p. 13.
- [32] R. Bredereck, J. Chen, and G. J. Woeginger. A characterization of the single-crossing domain. *Social Choice and Welfare*, 41(4):989–998, 2013. → p. 12.
- [33] L. Bulteau and J. Chen. d -dimensional Euclidean preferences. In preparation. → p. 12.
- [34] L. Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003. → p. 13.
- [35] L. Cai, J. Chen, R. G. Downey, and M. R. Fellows. Advice classes of parameterized tractability. *Annals of Pure and Applied Logic*, 84(1):119–138, 1997. → p. 3.
- [36] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. Combinatorial voter control in elections. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS’14)*, 2014. → p. 8.
- [37] J. Chen and J. Meng. On parameterized intractability: Hardness and completeness. *The Computer Journal*, 51(1):39–59, 2008. → p. 3.
- [38] J. Chen, K. Pruhs, and G. J. Woeginger. Characterizations of the one-dimensional Euclidean domain. In preparation. → p. 12.
- [39] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice. In *Proceedings of the 39th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM’07)*, volume 4362 of *LNCS*, pages 51–69. Springer, 2007. → p. 1.
- [40] V. Conitzer. Making decisions based on the preferences of multiple agents. *Communications of the ACM*, 53(3):84–94, 2010. → p. 1.

- [41] D. Cornaz, L. Galand, and O. Spanjaard. Bounded single-peaked width and proportional representation. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12)*, pages 270–275, 2012. → p. 13.
- [42] D. Cornaz, L. Galand, and O. Spanjaard. Kemeny elections with bounded single-peaked or single-crossing width. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, pages 76–82. AAAI Press, 2013. → p. 13.
- [43] M. J. A. N. C. de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris: L'Imprimerie Royale, 1785. → p. 2.
- [44] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2012. → p. 13.
- [45] C. Dodgson. A method of taking votes on more than two issues. Pamphlet printed by the Clarendon Press, Oxford, and headed “not yet published”, 1876. → p. 3.
- [46] B. Dorn and I. Schlotter. Multivariate complexity analysis of swap bribery. *Algorithmica*, 64(1):126–151, 2012. → pp. 5 and 7.
- [47] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer Verlag, 1999. → p. 2.
- [48] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer-Verlag, 2013. → pp. 3, 13, and 14.
- [49] R. G. Downey and D. M. Thilikos. Confronting intractability via parameters. *Computer Science Review*, 5(4):279–317, 2011. → p. 3.
- [50] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International World Wide Web Conference (WWW'01)*, pages 613–622. ACM, 2001. → p. 10.
- [51] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation revisited, 2001. Manuscript. → p. 10.
- [52] E. Elkind and P. Faliszewski. Approximation algorithms for campaign management. In *Proceedings of the 6th International Workshop on Internet and Network Economics (WINE'10)*, volume 6484 of *LNCS*, pages 473–482. Springer, 2010. → pp. 6 and 7.
- [53] E. Elkind, P. Faliszewski, and P. Skowron. A characterization of the single-peaked single-crossing domain. In *Proceedings of the 28th Conference on Artificial Intelligence (AAAI'14)*, 2014. → p. 12.
- [54] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory (SAGT'09)*, volume 5814 of *LNCS*, pages 299–310. Springer, 2009. → p. 6.
- [55] E. Elkind, P. Faliszewski, and A. M. Slinko. Clone structures in voters' preferences. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC'12)*, pages 496–513. ACM, 2012. → p. 13.

- [56] G. Erdélyi, M. Lackner, and A. Pfandler. Computational aspects of nearly single-peaked electorates. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI'13)*, pages 283–289. AAAI Press, 2013. → p. [13](#).
- [57] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009. → p. [6](#).
- [58] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010. → p. [1](#).
- [59] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. The complexity of manipulative attacks in nearly single-peaked electorates. *Artificial Intelligence*, 207:69–99, 2014. → p. [13](#).
- [60] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009. → p. [5](#).
- [61] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation*, 209(2):89–107, 2011. → p. [12](#).
- [62] P. Faliszewski and J. Rothe. Control and bribery. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*. Cambridge University Press. To appear. → p. [6](#).
- [63] M. R. Fellows, J. Flum, D. Hermelin, M. Müller, and F. A. Rosamond. W-hierarchies defined by symmetric gates. *Theory of Computing Systems*, 46(2):311–339, 2010. → p. [15](#).
- [64] M. R. Fellows, B. M. P. Jansen, and F. A. Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *European Journal of Combinatorics*, 34(3):541–566, 2013. → p. [16](#).
- [65] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer Verlag, 2006. → p. [3](#).
- [66] A. Frank and É. Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987. → p. [4](#).
- [67] M. R. Garey and D. S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979. → pp. [3](#) and [7](#).
- [68] L. A. Goodman. On methods of amalgamation. In R. M. Thrall, C. H. Coombs, and R. L. Davis, editors, *Decision Processes*, pages 39–48. John Wiley and Sons, Inc., 1954. → p. [3](#).
- [69] J. Guo, F. Hüffner, and R. Niedermeier. A structural view on parameterizing problems: Distance from triviality. In *Proceedings of the 1st International Workshop on Parameterized and Exact Computation (IWPEC'04)*, volume 3162 of *LNCS*, pages 162–173. Springer, 2004. → p. [13](#).
- [70] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007. → pp. [3](#) and [8](#).

- [71] J. Guo, R. Niedermeier, and S. Wernicke. Parameterized complexity of Vertex Cover variants. *Theory of Computing Systems*, 41(3):501–520, 2007. → p. 7.
- [72] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007. → p. 8.
- [73] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll’s 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997. → pp. 5 and 11.
- [74] R. Kannan. Minkowski’s convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987. → p. 4.
- [75] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):571–591, 1959. → p. 3.
- [76] V. Knoblauch. Recognizing one-dimensional Euclidean preference profiles. *Journal of Mathematical Economics*, 46(1):1–5, 2010. → p. 12.
- [77] D. Kurokawa, J. K. Lai, and A. D. Procaccia. How to cut a cake before the party ends. In *Proceedings of the 27th Conference on Artificial Intelligence (AAAI’13)*, pages 555–561. AAAI Press, 2013. → p. 16.
- [78] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983. → p. 4.
- [79] A. Levenglick. Fair and reasonable election systems. *Behavioral Science*, 20(1):34–46, 1975. → p. 3.
- [80] A. Lin. The complexity of manipulating k -approval elections. In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence (ICAART’11)*, pages 212–218. SciTePress, 2011. → p. 8.
- [81] H. Liu, H. Feng, D. Zhu, and J. Luan. Parameterized computational complexity of control problems in voting systems. *Theoretical Computer Science*, 410(27–29):2746–2753, 2009. → p. 8.
- [82] H. Liu and D. Zhu. Parameterized complexity of control problems in maximin election. *Information Processing Letters*, 110(10):383–388, 2010. → p. 8.
- [83] H. Liu and D. Zhu. Parameterized complexity of control by voter selection in Maximin, Copeland, Borda, Bucklin, and Approval election systems. *Theoretical Computer Science*, 498:115–123, 2013. → p. 9.
- [84] D. Lokshantov, N. Misra, and S. Saurabh. Kernelization–Preprocessing with a guarantee. In *The Multivariate Algorithmic Revolution and Beyond*, pages 129–161, 2012. → pp. 3 and 8.
- [85] D. Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008. → p. 7.

- [86] J. C. McCabe-Dansted. Approximability and computational feasibility of Dodgson’s rule. Master’s thesis, University of Auckland, 2006. → p. 4.
- [87] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, February 2006. → pp. 3, 4, and 13.
- [88] R. Niedermeier. Reflections on multivariate algorithmics and problem parameterization. In *Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS’10)*, volume 5 of *LIPIcs*, pages 17–32. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2010. → p. 16.
- [89] C. H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the V-C dimension. *Journal of Computer and System Sciences*, 53(2):161–170, 1996. → p. 14.
- [90] K. W. Roberts. Voting over income tax schedules. *Journal of Public Economics*, 8(3):329–340, 1977. → p. 12.
- [91] J. Robertson and W. Webb. *Cake-Cutting Algorithms—Be Fair If You Can*. A.K. Peters, Massachusetts, 1998. → pp. 1 and 15.
- [92] I. Schlotter, P. Faliszewski, and E. Elkind. Campaign management under approval-driven voting rules. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI’11)*, pages 726–731. AAAI Press, 2011. FPT approximation schemes available in an unpublished full version of the paper. → p. 8.
- [93] N. Simjour. *Parameterized Enumeration of Neighbour Strings and Kemeny Aggregations*. PhD thesis, University of Waterloo., 2013. → p. 11.
- [94] P. Skowron and P. Faliszewski. Approximating the maxcover problem with bounded frequencies in FPT time. Technical Report arXiv:1309.4405 [cs.DS], arXiv.org, Sept. 2013. → p. 7.
- [95] T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI’07)*, pages 3–8. AAAI Press, 2007. → p. 12.
- [96] J. Wang, M. Yang, J. Guo, Q. Feng, and J. Chen. Parameterized complexity of control and bribery for d -approval elections. In *Combinatorial Optimization and Applications - 7th International Conference (COCOA’13)*, volume 8287 of *LNCS*, pages 260–271. Springer, 2013. → p. 9.
- [97] Y. Yang and J. Guo. Complexity of control behaviors in k -peaked elections for a variant of approval voting. Manuscript, Max-Planck Institute, June 2013. arXiv:1304.4471v3 [cs.GT]. → p. 12.